material, subjective equivalence with NTSC is achieved. Also called standard digital television. See also *conventional definition television* and ITU-R Recommendation 1125.

start codes: 32-bit codes embedded in the coded bit stream that are unique. They are used for several purposes including identifying some of the layers in the coding syntax. Start codes consist of a 24 bit prefix (0x000001) and an 8 bit stream_id.

STD input buffer: A first-in, first-out buffer at the input of a system target decoder for storage of compressed data from elementary streams before decoding.

STD: See system target decoder.

still picture: A coded still picture consists of a video sequence containing exactly one coded picture which is intra-coded. This picture has an associated PTS and the presentation time of succeeding pictures, if any, is later than that of the still picture by at least two picture periods.

system clock reference (SCR): A time stamp in the program stream from which decoder timing is derived.

system header: The system header is a data structure that carries information summarizing the system characteristics of the Digital Television Standard multiplexed bit stream.

system target decoder (STD): A hypothetical reference model of a decoding process used to describe the semantics of the Digital Television Standard multiplexed bit stream.

time-stamp: A term that indicates the time of a specific action such as the arrival of a byte or the presentation of a presentation unit.

TOV: Threshold of visibility.

transport stream packet header: The leading fields in a transport stream packet up to and including the continuity_counter field.

variable bit rate: Operation where the bit rate varies with time during the decoding of a compressed bit stream.

VBV: See video buffering verifier.

Video buffering verifier (VBV): A hypothetical decoder that is conceptually connected to the output of an encoder. Its purpose is to provide a constraint on the variability of the data rate that an encoder can produce.

video sequence: A video sequence is represented by a sequence header, one or more groups of pictures, and an end_of_sequence code in the data stream.

8 VSB: Vestigial sideband modulation with 8 discrete amplitude levels.

16 VSB: Vestigial sideband modulation with 16 discrete amplitude levels.

3.4 Symbols, abbreviations, and mathematical operators

3.4.1 Introduction

The symbols, abbreviations, and mathematical operators used to describe the Digital Television Standard are those adopted for use in describing MPEG-2 and are similar to those used in the "C" programming language. However, integer division with truncation and rounding are specifically defined. The bitwise operators are defined assuming two's-complement representation of integers. Numbering and counting loops generally begin from 0.

3.4.2 Arithmetic operators

+	Addition.
-	Subtraction (as a binary operator) or negation (as a unary operator).
++	Increment.
- -	Decrement.
* or ×	Multiplication.
^	Power.
/	Integer division with truncation of the result toward 0. For example, 7/4 and -7/-4 are truncated to 1 and -7/4 and 7/-4 are truncated to -1.
//	Integer division with rounding to the nearest integer. Half-integer values are rounded away from 0 unless otherwise specified. For example 3//2 is rounded to 2, and -3//2 is rounded to -2.
DIV	Integer division with truncation of the result towards -∞.
%	Modulus operator. Defined only for positive numbers.
Sign( )	Sign(x) = 1 $x > 0$ 0 $x == 0$ -1 $x < 0$
NINT ( )	Nearest integer operator. Returns the nearest integer value to the real-valued argument. Half-integer values are rounded away from 0.
sin	Sine.
cos	Cosine.
exp	Exponential.
√	Square root.
$\log_{10}$	Logarithm to base ten.
$\log_e$	Logarithm to base e.

3.4.3 Logical operators

||	Logical OR.
&&	Logical AND.
!	Logical NOT.

3.4.4 Relational operators

>	Greater than.
≥	Greater than or equal to.
<	Less than.
≤	Less than or equal to.
==	Equal to.
!=	Not equal to.
max [,...,]	The maximum value in the argument list.
min [,...,]	The minimum value in the argument list.

3.4.5 Bitwise operators

&	AND.
|	OR.
>>	Shift right with sign extension.
<<	Shift left with 0 fill.

3.4.6 Assignment

=	Assignment operator.

3.4.7 Mnemonics

The following mnemonics are defined to describe the different data types used in the coded bit stream.

bslbf Bit string, left bit first, where "left" is the order in which bit strings are written in the Standard. Bit strings are written as a string of 1s and 0s within single quote marks, e.g. '1000 0001'. Blanks within a bit string are for ease of reading and have no significance.

uimsbf Unsigned integer, most significant bit first.

The byte order of multi-byte words is most significant byte first.

3.4.8 Constants

π	3.14159265359...

e 2.71828182845...

3.4.9 Method of describing bit stream syntax

Each data item in the coded bit stream described below is in bold type. It is described by its name, its length in bits, and a mnemonic for its type and order of transmission.

The action caused by a decoded data element in a bit stream depends on the value of that data element and on data elements previously decoded. The decoding of the data elements and definition of the state variables used in their decoding are described in the clauses containing the semantic description of the syntax. The following constructs are used to express the conditions when data elements are present, and are in normal type.

Note this syntax uses the "C" code convention that a variable or expression evaluating to a non-zero value is equivalent to a condition that is true.

```
while ( condition ) {
    data_element
    . . .
}
```

If the condition is true, then the group of data elements occurs next in the data stream. This repeats until the condition is not true.

```
do {
    data_element
    . . . }
 while ( condition )
```

The data element always occurs at least once. The data element is repeated until the condition is not true.

```
if ( condition) {
    data_element
    . . .
}
```

If the condition is true, then the first group of data elements occurs next in the data stream.

```
else {
    data_element
    . . .
}
```

If the condition is not true, then the second group of data elements occurs next in the data stream.

```
for (i = 0;i<n;i++) {
    data_element
    . . .
}
```

The group of data elements occurs n times. Conditional constructs within the group of data elements may depend on the value of the loop control variable i, which is set to zero for the first occurrence, incremented to 1 for the second occurrence, and so forth.

As noted, the group of data elements may contain nested conditional constructs. For compactness, the {} are omitted when only one data element follows.

data_element [ ] data_element [ ] is an array of data. The number of data elements is indicated by the context.

data_element [n] data_element [n] is the n+1th element of an array of data.

data_element [m][n] data_element [m][n] is the m+1,n+1 th element of a two-dimensional array of data.

data_element [l][m][n] data_element [l][m][n] is the l+1,m+1,n+1 th element of a three-dimensional array of data.

data_element [m..n] data_element [m..n] is the inclusive range of bits between bit m and bit n in the data_element.

Decoders must include a means to look for start codes and sync bytes (transport stream) in order to begin decoding correctly, and to identify errors, erasures or insertions while decoding. The methods to identify these situations, and the actions to be taken, are not standardized.

3.4.9.1 Definition of bytealigned function

The function bytealigned( ) returns 1 if the current position is on a byte boundary; that is, the next bit in the bit stream is the first bit in a byte. Otherwise it returns 0.

3.4.9.2 Definition of nextbits function

The function nextbits( ) permits comparison of a bit string with the next bits to be decoded in the bit stream.

3.4.9.3 Definition of next_start_code function

The next_start_code( ) function removes any zero bit and zero byte stuffing and locates the next start code.

This function checks whether the current position is byte-aligned. If it is not, 0 stuffing bits are present. After that any number of 0 bytes may be present before the start-code. Therefore start-codes are always byte-aligned and may be preceded by any number of 0 stuffing bits.

Table 3.1 Next Start Code

Syntax	No. of bits	Mnemonic
next_start_code( ) {		
while (!bytealigned( ))		
zero_bit	1	'0'
while (nextbits( )!='0000 0000 0000 0000 0000 0001')		
zero_byte	8	'00000000'
}		

4. SYSTEM OVERVIEW

4.1 *Objectives*

The Digital Television Standard describes a system designed to transmit high quality video and audio and ancillary data over a single 6 MHz channel. The system can deliver reliably about 19 Mbps of throughput in a 6 MHz terrestrial broadcasting channel and about 38 Mbps of throughput in a 6 MHz cable television channel. This means that encoding a video source whose resolution can be as high as five times that of conventional television (NTSC) resolution requires a bit rate reduction by a factor of 50 or higher. To achieve this bit rate reduction, the system is designed to be efficient in utilizing available channel capacity by exploiting complex video and audio compression technology.

The objective is to maximize the information passed through the data channel by minimizing the amount of data required to represent the video image sequence and its associated audio. The objective is to represent the video, audio, and data sources with as few bits as possible while preserving the level of quality required for the given application.

Although the RF/Transmission subsystems described in the Digital Television Standard are designed specifically for terrestrial and cable applications, the objective is that the video, audio, and service multiplex/transport subsystems be useful in other applications.

4.2 *System block diagram*

A basic block diagram representation of the system is shown in Figure 4.1. This representation is based on one adopted by the International Telecommunication Union, Radiocommunication Sector (ITU-R), Task Group 11/3 (Digital Terrestrial Television Broadcasting). According to this model, the digital television system can be seen to consist of three subsystems.[1]

1. Source coding and compression,
2. Service multiplex and transport, and
3. RF/Transmission.

"Source coding and compression" refers to the bit rate reduction methods, also known as data compression, appropriate for application to the video, audio, and ancillary digital data streams. The term "ancillary data" includes control data, conditional access control data, and data associated with the program audio and video services, such as closed captioning. "Ancillary data" can also refer to independent program services. The purpose of the coder is to minimize the number of bits needed to represent the audio and video information. The digital television system employs the MPEG-2 video stream syntax for the coding of video and the Digital Audio Compression (AC-3) Standard for the coding of audio.

[1] ITU-R Document TG11/3-2, "Outline of Work for Task Group 11/3, Digital Terrestrial Television Broadcasting," 30 June 1992.

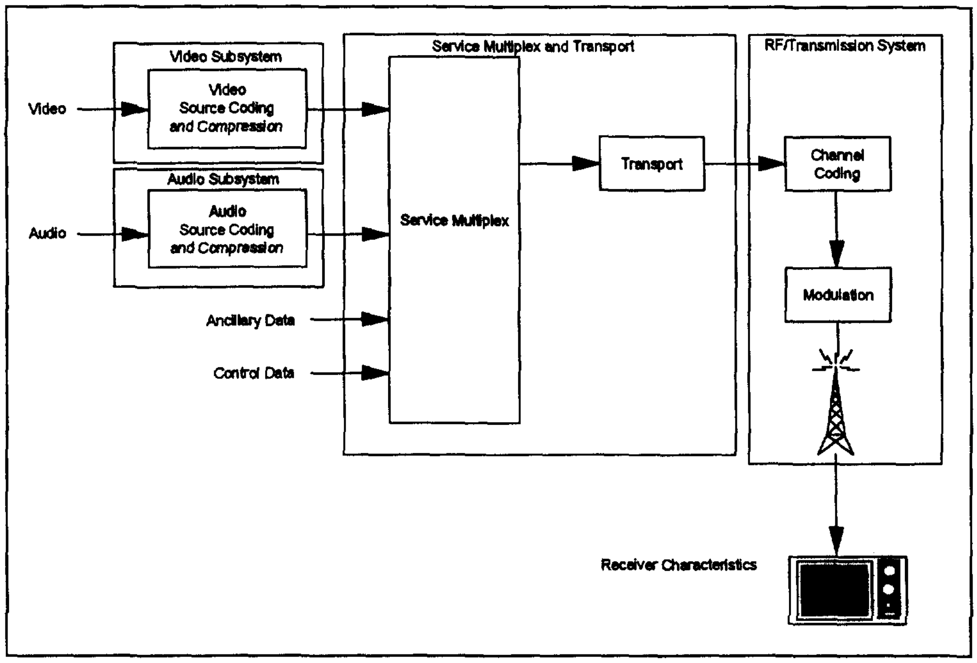


Figure 4.1. ITU-R digital terrestrial television broadcasting model.

"Service multiplex and transport" refers to the means of dividing the digital data stream into "packets" of information, the means of uniquely identifying each packet or packet type, and the appropriate methods of multiplexing video data stream packets, audio data stream packets, and ancillary data stream packets into a single data stream. In developing the transport mechanism, interoperability among digital media, such as terrestrial broadcasting, cable distribution, satellite distribution, recording media, and computer interfaces, was a prime consideration. The digital television system employs the MPEG-2 transport stream syntax for the packetization and multiplexing of video, audio, and data signals for digital broadcasting systems.[2] The MPEG-2 transport stream syntax was developed for applications where channel bandwidth or recording media capacity is limited and the requirement for an efficient transport mechanism is paramount. It was designed also to facilitate interoperability with the ATM transport mechanism.

"RF/Transmission" refers to channel coding and modulation. The channel coder takes the data bit stream and adds additional information that can be used by the receiver to reconstruct the data from the received signal which, due to transmission impairments, may not accurately represent the transmitted signal. The modulation (or physical layer) uses the digital data stream information to modulate the transmitted signal. The modulation subsystem utilizes an 8 VSB technique for terrestrial transmission and a 16 VSB technique for high data-rate cable delivery.

The Chapters that follow consider the characteristics of the subsystems necessary to accommodate the services envisioned.

[2] Chairman, ITU-R Task Group 11/3, "Report of the Second Meeting of ITU-R Task Group 11/3, Geneva, 13-19 October 1993," 5 January 1994, p. 40.

5. VIDEO SYSTEMS

5.1 Overview of video compression and decompression

The need for compression in a digital HDTV system is apparent from the fact that the bit rate required to represent an HDTV signal in uncompressed digital form is about 1 Gbps, and the bit rate that can reliably be transmitted within a standard 6 MHz television channel is about 20 Mbps. This implies a need for about a 50:1 or greater compression ratio.

The Digital Television Standard specifies video compression using a combination of compression techniques, and for reasons of compatibility these compression algorithms have been selected to conform to the specifications of MPEG-2, which is a flexible internationally accepted collection of compression algorithms.

The purpose of this tutorial exposition is to identify the significant processing stages in video compression and decompression, giving a clear explanation of what each processing step accomplishes, but without including all the details that would be needed to actually implement a real system. Those necessary details in every case are specified in the normative part of the standards documentation, which shall in all cases represent the most complete and accurate description of the video compression. Because the video coding system includes a specific subset of the MPEG-2 toolkit of algorithmic elements, another purpose of this tutorial is to clarify the relationship between this system and the more general MPEG-2 collection of algorithms.

5.1.1 MPEG-2 levels and profiles

The MPEG-2 specification is organized into a system of profiles and levels, so that applications can ensure interoperability by using equipment and processing that adhere to a common set of coding tools and parameters.[3] The Digital Television Standard is based on the MPEG-2 Main Profile at the High Level (MP@HL). The Main Profile includes three types of frames for prediction (I-frames, P-frames, and B-frames), and an organization of luminance and chrominance samples (designated 4:2:0) within the frame. The Main Profile does not include a scalable algorithm, where scalability implies that a subset of the compressed data can be decoded without decoding the entire data stream. The High Level includes formats with up to 1152 active lines and up to 1920 samples per active line, and for the Main Profile is limited to a compressed data rate of no more than 80 Mbps. The parameters specified by the Digital Television Standard represent specific choices within these constraints.

5.1.2 Compatibility with MPEG-2

The video compression system does not include algorithmic elements that fall outside the specifications for MPEG-2 Main Profile, High Level. Thus video decoders

[3] For more information about profiles and levels see ISO/IEC 13818-2, Section 8.

which conform to the MPEG-2 MP@HL can be expected to decode bit streams produced in accordance with the Digital Television Standard. Note that it is not necessarily the case that all video decoders which are based on the Digital Television Standard will be able to properly decode all video bit streams which comply to MPEG-2 MP@HL.

5.1.3 Overview of video compression

The video compression system takes in an analog video source signal and outputs a compressed digital signal that contains information that can be decoded to produce an approximate version of the original image sequence. The goal is for the reconstructed approximation to be imperceptibly different from the original for most viewers, for most images, for most of the time. In order to approach such fidelity, the algorithms are flexible, allowing for frequent adaptive changes in the algorithm depending on scene content, history of the processing, estimates of image complexity and perceptibility of distortions introduced by the compression.

Figure 5.1 shows the overall flow of signals in the ATV system. Note that analog signals presented to the system are digitized and sent to the encoder for compression, and the compressed data then are transmitted over a communications channel. On being received, the possibly error-corrupted compressed signal is decompressed in the decoder, and reconstructed for display.

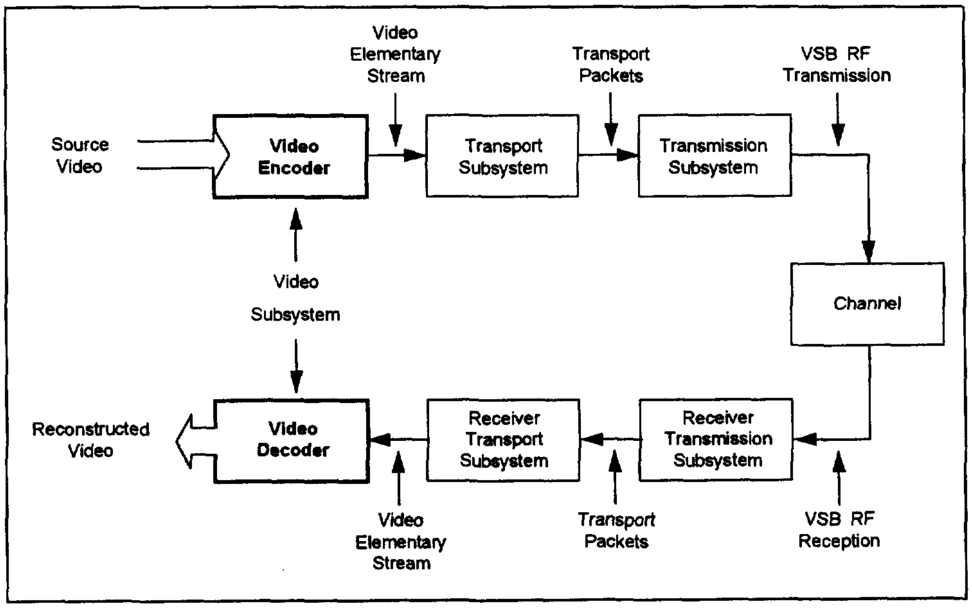


Figure 5.1. Video coding in relation to the ATV system.

5.2 Video preprocessing

Video preprocessing converts the analog input signals to digital samples in the form needed for the subsequent compression. The analog input signals are red (R), green (G), and blue (B) signals.

5.2.1 Sampling rates

For the 720 line format, with 750 total lines per frame and 1650 total samples per line, the sampling frequency will be 74.25 MHz for the 60.00 frames per second (fps) frame rate. For the 1080 line format, with 1125 total lines per frame and 2200 total samples per line, the sampling frequency will be 74.25 MHz for the 30.00 fps frame rate. Note that both 59.94 fps and 60.00 fps are acceptable as frame or field rates for the system.

5.2.2 Precision of samples

Samples are typically obtained using analog-to-digital converter circuits with 8-bit precision. After preprocessing, the various luminance and chrominance samples will typically be represented using 8 bits per sample of each component.

5.2.3 Source-adaptive processing

The image sequences that constitute the HDTV source signal can vary in spatial resolution (720 lines or 1080 lines) and in temporal resolution (60 fps, 30 fps, or 24 fps). The video compression system accommodates the differences in source material to maximize the efficiency of compression.

5.2.4 Film mode

When a large fraction of pixels do not change from one frame in the image sequence to the next, a video encoder may automatically recognize that the input was film with an underlying frame rate less than 60 fps.

In the case of 24 fps film material that is sent at 60 Hz using a 3:2 pull-down operation, the processor may detect the sequences of three nearly identical pictures followed by two nearly identical pictures, and only encode the 24 unique pictures per second that existed in the original film sequence. When 24 fps film is detected by observation of the 3:2 pull-down pattern, the input signal is converted back to a progressively scanned sequence of 24 frames per second prior to compression. This avoids sending redundant information, and allows the encoder to provide an improved quality of compression. The encoder indicates to the decoder that the film mode is active.

In the case of 30 fps film material that is sent at 60 Hz, the processor may detect the sequences of two nearly identical pictures followed by two nearly identical pictures. In that case, the input signal is converted back to a progressively scanned sequence of 30 frames per second.

5.2.5 Color component separation and processing

The input video source to the ATV video compression system is in the form of RGB components matrixed into luminance (Y) and chrominance (Cb and Cr) components using a linear transformation (3-by-3 matrix, specified in the standard). The luminance component represents the intensity, or black-and-white picture, while the chrominance components contain color information. The original RGB components are highly correlated with each other; the resulting Y, Cb, and Cr signals have less correlation and are thus easier to code efficiently; The luminance and chrominance components correspond to functioning of the biological vision system; that is, the human visual system responds differently to the luminance and chrominance components.

The coding process may take advantage also of the differences in the ways that humans perceive luminance and chrominance. In the Y, Cb, Cr color space, most of the high frequencies are concentrated in the Y component; the human visual system is less sensitive to high frequencies in the chrominance components than to high frequencies in the luminance component. To exploit these characteristics the chrominance components are low-passed filtered in the ATV video compression system and sub-sampled by a factor of two along both the horizontal and vertical dimensions, producing chrominance components that are one-fourth the spatial resolution of the luminance component.

5.2.6 Anti-alias filtering

The Y, Cb, and Cr components are applied to appropriate low-pass filters that shape the frequency response of each of the three components. Prior to horizontal and vertical sub-sampling of the two chrominance components, they may be processed by half-band filters in order to prevent aliasing.[4]

5.2.7 Number of lines encoded

The video coding system requires that the coded picture area has a number of lines that is a multiple of 32 for an interlaced format, and a multiple of 16 for a non-interlaced format. This means that for encoding the 1080-line format, a coder must actually deal with 1088 lines (1088 = 32 x 34). The extra eight lines are in effect “dummy” lines having no content, and the coder designers will choose dummy data that simplifies the implementation. The extra eight lines are always the last eight lines of the encoded image. These dummy lines do not carry useful information, but add little to the data required for transmission.

5.3 Representation of picture data

Digital television uses a digital representation of the image data, which allows the data to be processed using computer-like digital processing. The process of digitization involves sampling of the analog television signals and their components, and representing each sample with a digital code.

5.3.1 Pixels

The analog video signals are sampled in a sequence that corresponds to the scanning raster of the television format; *i.e.*, from left to right within a line, and in lines from top to bottom. The collection of samples in a single frame, or in a single field for

[4] For more information about aliasing and sampling theory, see James A. Cadzow, *Discrete Time Systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

interlaced images, is treated together, as if they all corresponded to a single point in time (in the case of film modes, they do in fact correspond to a single time or exposure interval). The individual samples of image data are referred to as picture elements, or "pixels," or "pels." A single frame or field can then be thought of as a rectangular array of pixels.

5.3.1.1 Square pixels

When the ratio of active pixels per line to active lines per frame is the same as the display aspect ratio, which is 16:9, the format is said to have "square" pixels. The term refers to the spacing of samples and does not refer to the shape of the pixel, which might ideally be a point with zero area from a mathematical sampling point of view.

5.3.1.2 Spatial relationship between luminance and chrominance samples

As described in Section 5.2.5, the chrominance component samples are sub-sampled by a factor of two in both horizontal and vertical directions. This means the chrominance samples are spaced twice as far apart as the luminance samples, and it is necessary to specify the location of chrominance samples relative to the luminance samples.

Figure 5.2 illustrates the spatial relationship between chrominance and luminance samples. For every four luminance samples, there are one each of the Cb and Cr chroma samples. The Cb and Cr chroma samples are located in the same place.

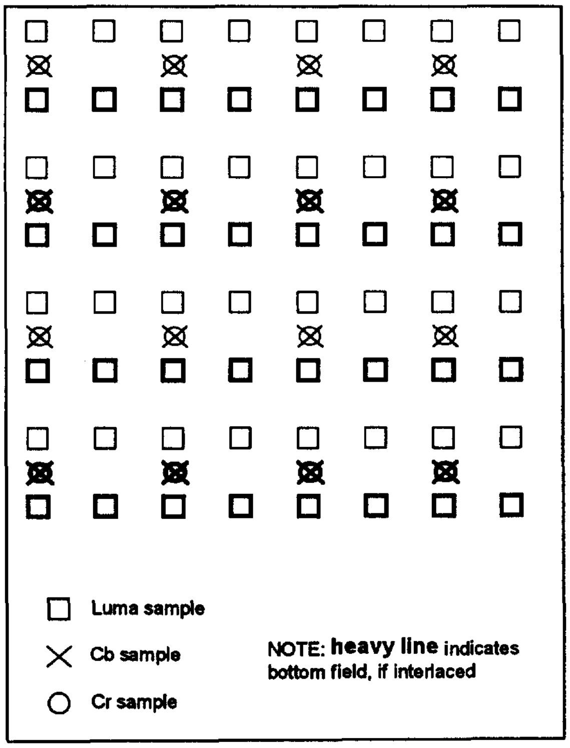


Figure 5.2. Placement of luma/chroma samples for 4:2:0.

Note that the vertical spatial location of chrominance samples does not correspond to an original sample point, but lies halfway between samples on two successive lines. The 4:2:0 sampling structure thus requires the Cb and Cr samples to be interpolated. For progressively scanned source pictures the processor may simply average the two adjacent (upper and lower) values to compute the sub-sampled values.

In the case of interlaced pictures, it can be seen in Figure 5.2 that the vertical positions of the chrominance samples in a field are not halfway between the luminance samples of the same field. This is done so that the spatial locations of the chrominance samples in the frame are the same for both interlaced and progressive sources.

5.3.2 Blocks of pixels

The pixels are organized into blocks for the purpose of further processing. A block consists of an array of pixel values or an array that is some transform of pixel values. A block for the ATV system is defined as an array of 8-by-8 values representing either luminance or chrominance information (see Figure 5.3a).

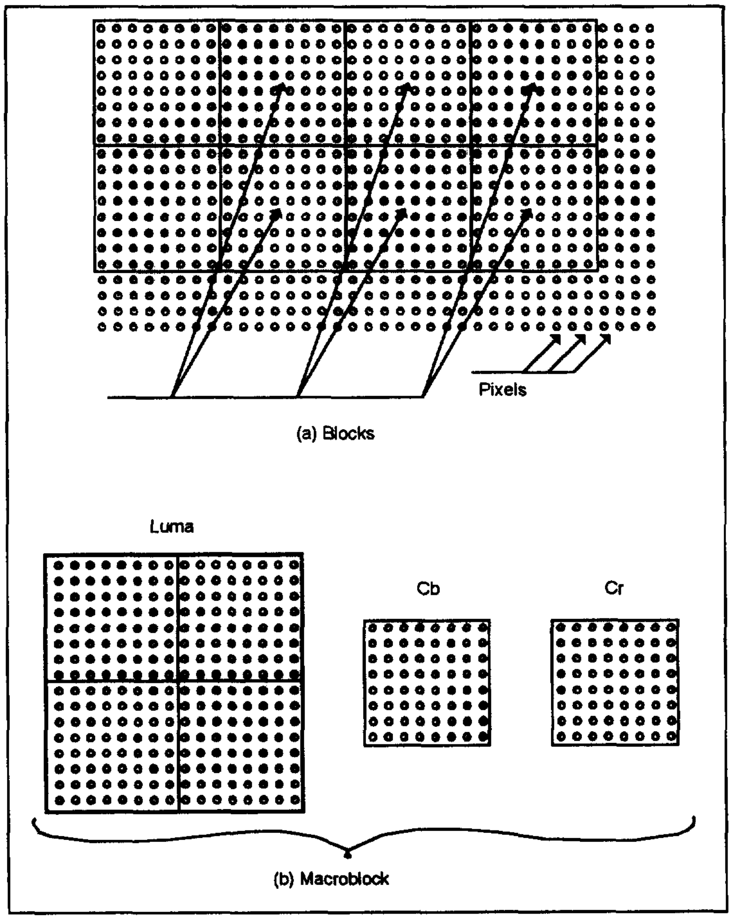


Figure 5.3. Blocks and macroblocks.

5.3.3 Macroblocks

Blocks of information are organized into macroblocks. A macroblock consists of four blocks of luminance (or a 16 pixel by 16 line region of values) and two chroma (Cb and Cr) blocks. The term *macroblock* may be used to refer directly to pel data or to the transformed and coded representation of pel data. As shown in Figure 5.3b, this yields 256 luminance samples and 64 Cb samples and 64 Cr samples (total of 384) per macroblock.

For the 720 line format (with 1280 samples per line), there are therefore 45 rows of macroblocks, with 80 macroblocks per row. For the 1080-line format (with 1920 samples per line), there are 68 rows of macroblocks (including the last row that adds eight dummy lines to create the 1088 lines for coding), with 120 macroblocks per row.

5.3.4 Slices

One or more contiguous macroblocks within the same row are grouped together to form slices. The order of the macroblocks within a slice is the same as the conventional television raster scan being from left to right.

Slices provide a convenient mechanism for limiting the propagation of errors. Since the coded bit stream consists mostly of variable-length codewords, any uncorrected transmission errors will cause a decoder to lose its sense of codeword alignment. Each slice begins with a slice start code. Since the MPEG codeword assignment guarantees that no legal combination of codewords can emulate a start code, the slice start code can be used to regain the sense of codeword alignment after an error. When an error occurs in the data stream, the decoder can thus skip to the start of the next slice and resume correct decoding.

The number of slices affects the compression efficiency; partitioning the data stream to have more slices provides for better error recovery but uses bits that could otherwise be used to improve picture quality. The slice is the minimum unit for resynchronization after an error.

In the ATV system, the initial macroblock of every horizontal row of macroblocks is also the beginning of a slice, with possibly several slices across the row.

5.3.5 Pictures, groups of pictures, and sequences

The primary coding unit of a video sequence is the individual video frame or picture. A video picture consists of the collection of slices which constitute the active picture area.

A video sequence consists of a collection of one or more consecutive pictures. A video sequence commences with a sequence header and is terminated by an end-of-sequence code in the data stream. A video sequence can contain additional sequence headers. Any video sequence header can serve as an entry point. An entry point is a point in the coded video bit stream after which a decoder can become properly initialized and correctly parse the bit stream syntax.

One or more pictures (frames) in sequence may be combined into a Group of Pictures (GOP) to provide boundaries for inter-picture coding and registration of time code. GOPs are optional within both MPEG-2 and the ATV system.

Figure 5.4 illustrates a time sequence of video frames consisting of intra-coded pictures (I-frames); predictive coded pictures (P-frames), and bidirectionally predictive coded pictures (B-frames).

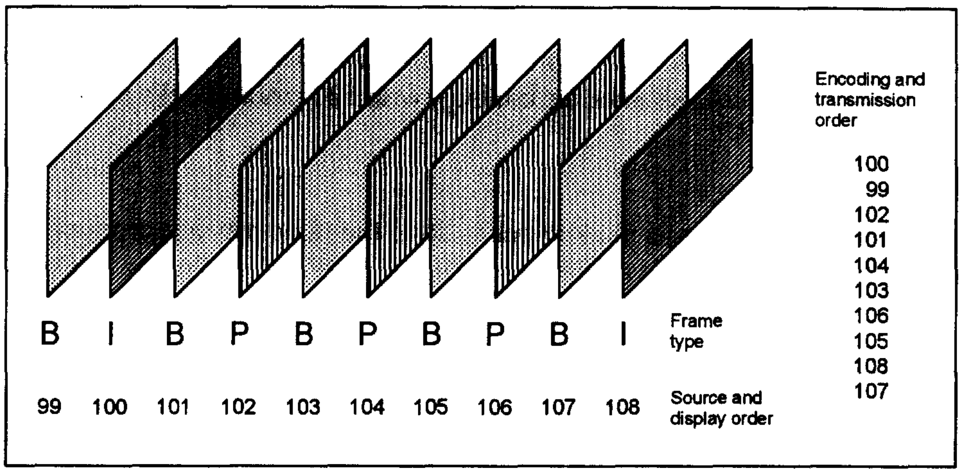


Figure 5.4. Illustration of I-frames, P-frames, and B-frames.

5.3.5.1 I-frames

Some elements of the compression process exploit only the spatial redundancy within a single picture (frame or field). These processes are called intraframe coding, and do not take advantage of the temporal correlation addressed by temporal prediction, which is referred to as interframe coding. Frames that do not use any interframe coding are referred to as I-frames (where "I" denotes *intraframe* coded). The ATV video compression system utilizes both intraframe coding and interframe coding.

The use of periodic I-frames facilitates receiver initializations and channel acquisition (when the receiver is turned on or the channel is changed). The decoder can take advantage of the intraframe coding mode when non-correctable channel errors occur. With motion-compensated prediction, an initial frame must be available at the decoder to start the prediction loop. Therefore, a mechanism must be built into the system so that if the decoder loses synchronization for any reason, it can rapidly reacquire tracking.

The frequency of occurrence of I-pictures may vary and is selected at the encoder. This allows consideration to be given to the need for random access and the location of scene cuts in the video sequence.

5.3.5.2 P-frames

P-frames (where "P" denotes *predicted*) are frames where the temporal prediction is in the forward direction only (i.e., predictions for the P-frame are formed only from pixels in the most recently decoded I or P-frame). These forward-predicted frames allow the exploitation of interframe coding techniques to improve the overall compression efficiency and picture quality. P-frames may include portions that are only intraframe coded. Each macroblock within a P-frame can be either forward-predicted or intraframe coded.

5.3.5.3 B-frames

The B-frame (where "B" denotes *bidirectionally* predicted) is a picture type within the coded video sequence that includes prediction from a future frame as well as from a previous frame. The referenced future or previous frames, sometimes called "anchor" frames, are in all cases either I or P-frames.

The basis of the B-frame prediction is that a video frame is correlated both with frames which occur in the past and frames which occur in the future. Consequently, if a future frame is available to the decoder, a superior prediction can be formed, thus saving bits and improving performance. Some of the consequences of using future frames in the prediction are: the B-frame cannot be used for predicting future frames, the transmission order of frames is different from the displayed order of frames, and the encoder and decoder must reorder the video frames thereby increasing the total latency. In the example illustrated in Figure 5.4, there is one B-frame between each pair of I/P-frames. Each frame is labeled with both its display order and transmission order. The I and P frames are transmitted out of sequence so the video decoder has both anchor frames decoded and available for prediction.

B-frames are used for increasing the compression efficiency and perceived picture quality when encoding latency is not an important factor. The use of B-frames increases coding efficiency for both interlaced and progressively scanned material. B-frames are included in the ATV system because the increase in compression efficiency is noticeable especially with progressive scanning where techniques such as dual prime (see Section 5.5.2) are not available. The choice of number of bidirectional pictures between any pair of reference (I or P) frames can be determined at the encoder.

5.4 Motion estimation

As explained in Section 5.5, "Encoder prediction loop," the compression algorithm depends on creating an estimate of the image being compressed, and subtracting from the image to be compressed the pixel values of the estimate or prediction. If the estimate is good, the subtraction will leave a very small residue to be transmitted; in fact, if the estimate or prediction were perfect, the difference would be zero for all the pixels in the frame of differences, and no new information would need to be sent (that condition can be approached for still images).

If the estimate is not close to zero for some pixels or many pixels, those differences represent information that needs to be transmitted so the decoder can reconstruct a correct image. The kinds of image sequences that cause large prediction differences include severe motion and/or sharp details.

5.4.1 Vector search algorithm

The video coding system uses motion compensated prediction as part of the data compression process. Thus macroblocks in the current frame of interest are predicted by macroblock-sized regions in previously transmitted frames. Motion compensation refers to the fact that the locations of the macroblock-sized regions in the reference frame can be offset to account for local motions. The macroblock offsets are known as *motion vectors.*

This standard does not specify how encoders should determine motion vectors. One possible approach might be to perform an exhaustive search to determine the vertical and horizontal offsets that minimize the total difference between the offset region in the reference frame and the macroblock in the frame to be coded.

5.4.2 Motion vector precision

The estimation of interframe displacement is calculated with half-pel precision, in both vertical and horizontal dimensions. That means the displaced macroblock from the previous frame can be displaced by non-integer displacements, and will require interpolation to compute the values of displaced picture elements at locations not in the original array of samples. Estimates for half-pel locations are computed by averages of adjacent sample values.

5.4.3 Motion vector coding

Motion vectors within a slice are differenced, so that the first value for a motion vector is transmitted directly, and the following sequence of motion vectors differences is sent using variable-length codes (VLC).

5.4.4 Estimation at frame boundaries

Motion vectors are constrained so that all pixels from the motion compensated prediction region in the reference picture fall within the picture boundaries.

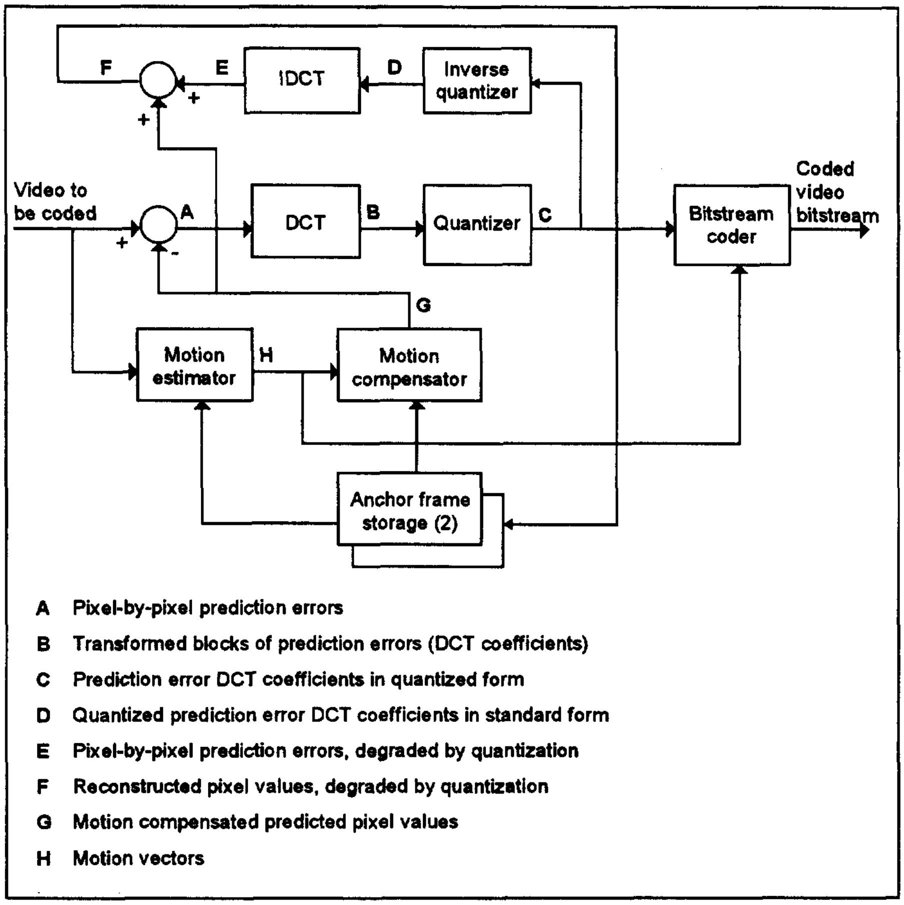
5.5 Encoder prediction loop

The best way to understand the way the different algorithmic elements combine to achieve video compression is to examine the encoder prediction loop. This closed feedback loop, shown in the simplified block diagram of Figure 5.5, is the heart of the video compression for the ATV system.

5.5.1 Prediction loop block diagram

The prediction loop contains a prediction function that estimates, or predicts, the picture values of the next picture to be encoded in the sequence of successive pictures that constitute the TV program. This prediction is based on previous information that is available within the loop, derived from earlier pictures. The transmission of the predicted compressed information works because the very same information used to make the prediction is available also at the receiving decoder (barring transmission errors, which are expected to be infrequent within the coverage area).

The subtraction of the predicted picture values from the new picture to be coded is at the core of predictive coding. The goal is to do such a good job of predicting the new values that the result of the subtraction function at the beginning of the prediction loop is zero or close to zero for most of the time.



A Pixel-by-pixel prediction errors
B Transformed blocks of prediction errors (DCT coefficients)
C Prediction error DCT coefficients in quantized form
D Quantized prediction error DCT coefficients in standard form
E Pixel-by-pixel prediction errors, degraded by quantization
F Reconstructed pixel values, degraded by quantization
G Motion compensated predicted pixel values
H Motion vectors

Figure 5.5. Encoder prediction loop.

The prediction differences are computed separately for the luminance and two chrominance components before further processing.

As discussed under I-frames, there are times when prediction is not used, for part of a frame or for an entire frame. Those portions are said to be "intraframe" coded, while the portions that use the prediction from previous or future pictures are said to be "interframe" coded.

5.5.1.1 Spatial transform block — DCT

The prediction differences (sometimes referred to as prediction errors) are grouped into 8-by-8 blocks and a spatial transform is applied to the blocks of difference values. In the intraframe case, the spatial transform is applied to the raw, undifferenced picture data. The luminance and two chrominance components are separately transformed. Since the chrominance data is sub-sampled vertically and horizontally, each 8-by-8 block of chrominance (Cb or Cb) data corresponds to a 16-by-16 macroblock of luminance data, which is not sub-sampled.

The spatial transform used is the discrete cosine transform, or DCT. The formula for transforming the data is given by:

$$F(u,v)=\frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

where x and y are pixel indices within an 8-by-8 block, u and v are DCT coefficient indices within an 8-by-8 block, and:

$$C(w)=\frac{1}{\sqrt{2}}\quad \textit{for } w=0$$

$$C(w)=1\quad \textit{for } w=1,2,\ldots,7$$

Thus an 8-by-8 array of numbers $f(x, y)$ is the input to a mathematical formula, and the output is an 8-by-8 array of different numbers, $F(u, v)$. The inverse transform[5] is given by:

$$f(x,y)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)F(u,v)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

In principle, applying the inverse DCT transform to the transformed array would yield exactly the same array as the original. In that sense, transforming the data doesn't modify the data but merely represents the data in a different form.

The decoder uses the inverse transformation to approximately reconstruct the arrays that were transformed at the encoder, as part of the process of decoding the received compressed data. The approximation in that reconstruction is controlled in advance during the encoding process so as to minimize the visual effects of coefficient inaccuracies, while reducing the quantity of data that needs to be transmitted.

DCT transforms are discussed further in Section 5.7.

5.5.1.2 Quantizer

The process of transforming the original data organizes the information in a way that exposes the spatial frequency components of the images or image differences. Using knowledge about the response of the human visual system to different spatial frequencies, the encoder can selectively adjust the precision of transform coefficient representation. The goal is to include as much information about a particular spatial frequency as needed (and possible, given constraints on data transmission), but not to use more precision than is needed, based on visual perception criteria.

[5] The IDCT is required to conform to *IEEE Standard Specifications for the Implementation of 8x8 Inverse Discrete Cosine Transform*, Std 1180-1990, December 6, 1990.

For example, in a portion of a picture that is very "busy" with much detail, imprecision in reconstructing spatial high frequency components in a small region might be masked by the picture's local "busy-ness." On the other hand, very precise representation and reconstruction of the average value or DC term of the DCT block (the F(0,0) term of the transformed coefficients represents the average of the original 64 coefficients and is referred to as the DC term) would be important in a smooth area of sky.

Recall that the DCT of each 8-by-8 block of pixel values produces an 8-by-8 array of DCT coefficients. The relative precision accorded to each of the 64 DCT coefficients can be selected according to its relative importance in human visual perception. The relative coefficient precision information is represented by a *quantizer matrix*, which is an 8-by-8 array of values. Each value in the quantizer matrix represents the coarseness of quantization of the related DCT coefficient.

Two types of quantizer matrices are supported — one which is used for macroblocks which are intraframe coded, and the other which is used for non-intraframe coded macroblocks. The video coding system defines default values for both the intra-quantizer and the non-intra-quantizer matrices. Either or both of the quantizer matrices can be overridden at the picture level by transmitting the appropriate arrays of 64 values. Any quantizer matrix overrides stay in effect until the following sequence start code.

The transform coefficients, which represent the bulk of the actual coded video information, are quantized to various degrees of coarseness. As indicated above, the appearance of some portions of the picture will be more affected than others to the loss of precision through coefficient quantization. This phenomenon is exploited by the availability of the quantizer scale factor, which allows the overall level of quantization to vary for each macroblock. Thus entire macroblocks which are deemed to be visually less important can be quantized more coarsely, which results in decreasing the number of bits needed to represent the picture.

For each coefficient other than the DC coefficient of intraframe coded blocks, the quantizer scale factor is multiplied by the corresponding value in the appropriate quantizer matrix to form the quantizer step size. Quantization of the DC coefficients of intra-coded blocks is unaffected by the quantizer scale factor, and is only governed by the (0, 0) element of the intra-quantizer matrix, which is always set to be 8 by ISO/IEC 13818-2.

Quantization is discussed further in Section 5.8.

5.5.1.3 Entropy coder

An important effect of the quantization of transform coefficients is that many coefficients will be rounded to zero after quantization. In fact, a primary method of controlling the encoded data rate is the control of quantization coarseness, since a coarser quantization leads to an increase in the number of zero-value quantized coefficients. Entropy coding is discussed in greater detail in Section 5.9.

5.5.1.4 Inverse quantizer

At the decoder the coded coefficients are decoded and an 8-by-8 block of quantized coefficients is reconstructed. Each of these 64 coefficients is *inverse quantized* according to the prevailing quantizer matrix, quantizer scale, and frame type. The result of inverse quantization is a block of 64 DCT coefficients.

5.5.1.5 Inverse spatial transform block — IDCT

The decoded and inverse quantized coefficients are organized as 8-by-8 blocks of DCT coefficients and the inverse discrete cosine transform (IDCT) is applied to each block. This results in a new array of pixel values, or pixel difference values that correspond to the output of the subtraction at the beginning of the prediction loop. If the prediction loop was in the interframe mode, the values will be pixel differences. If the mode was in the intraframe mode, then the inverse transform will produce pixel values directly.

5.5.1.6 Motion compensator

If a portion of the image has not moved, then it is easy to see that a subtraction of the old portion from the new portion of the image will produce zero or nearly zero pixel differences, which is the goal of the prediction.

If there has been movement in the portion of the image under consideration, the direct pixel-by-pixel differences will in general not be zero, and might be statistically very large. However, the motion in most natural scenes is organized, and can in most cases be approximately represented locally as a translation. For this reason the video coding system allows for *motion compensated* prediction, whereby macroblock sized regions in the reference frame may be translated vertically and horizontally with respect to the macroblock being predicted, to compensate for local motion.

The pixel-by-pixel differences between the current macroblock and the motion compensated prediction are transformed by the DCT and quantized using the composition of the non-intra-quantizer matrix and the quantizer scale factor. The quantized coefficients are then coded.

5.5.1.7 Anchor frames

In the case of I-frames, the entire frame is encoded without reference to any other coded frames. P-frames are referenced to the most recently decoded I or P-frame). B-frames, however, permit the use of two frames as prediction references. One of the reference frames occurs earlier than the coded frame in display order (which can be used for forward prediction), and the other occurs later in display order (which can be used for backward prediction).

For a given macroblock within the B-frame, the encoder has four options. They are: forward prediction, backward prediction, bidirectional prediction, and intraframe coding. When bidirectional prediction is used, the forward and backward predictors are averaged and then subtracted from the target macroblock to form the prediction error.

The prediction error is then transformed, quantized and transmitted in the usual manner. Note that both of the frames used as references in coding a B-frame are coded and transmitted prior to the coding of the actual B-frame. This results in the need for frame reordering within the decoder to produce the proper display order.

5.5.2 Dual prime prediction mode

The dual prime prediction mode is an alternative "special" prediction mode which is based on field-based motion prediction but requires fewer transmitted motion vectors than conventional field-based prediction. This mode of prediction is available only for interlaced material and only when the encoder configuration does not use B-frames. This mode of prediction may be particularly useful for improving encoder efficiency for low delay applications.

The basis of dual prime prediction is that field-based predictions of both fields in a macroblock are obtained by averaging two separate predictions which are predicted from the two nearest decoded fields in time. Each of the macroblock fields is predicted separately, although the four vectors (one pair per field) used for prediction are all derived from a single transmitted field-based motion vector. In addition to the single field-based motion vector, a small "differential" vector (limited to vertical and horizontal component values of +1, 0 and -1) is also transmitted for each macroblock. Together, these vectors are used to calculate the pairs of motion vectors for each macroblock. The first prediction in the pair is simply the transmitted field-based motion vector. The second prediction vector is obtained by combining the differential vector with a scaled version of the first vector. Once both predictions are obtained, a single prediction for each macroblock field is calculated simply by averaging each pel in the two original predictions. The final averaged prediction is then subtracted from the macroblock field being encoded.

5.5.3 Adaptive field/frame prediction mode

Interlaced pictures may be coded in either of two ways — either as two separate fields or as a single frame. When coded as separate fields, all of the codes for the first field are transmitted as a unit before the codes for the second field. When coded as a frame, information for both fields is coded for each macroblock.

When frame-based coding is used with interlaced pictures, each macroblock may be selectively coded using either field prediction or frame prediction. When frame prediction is used, a motion vector is applied to a picture region which is made up of both parity fields interleaved together. When field prediction is used, a motion vector is applied to a region made up of scan lines from a single field. Field prediction allows the selection of either parity field to be used as a reference for the field being predicted.

5.6 Image refresh

As described in the preceding sections, a given picture may be sent by describing the differences between it and one or two previously transmitted pictures. In order for this scheme to work, there must be some way for decoders to become initialized with a valid picture upon tuning into a new channel, or to become re-initialized with a valid picture after experiencing transmission errors. Additionally, it is necessary to limit the number of consecutive predictions which can be performed in a decoder to control the buildup of errors due to *IDCT mismatch*.

IDCT mismatch arises from the fact that the video coding system, by intention, does not completely specify the results of the IDCT operation.[6] It is thus possible for the reconstructed pictures in a decoder to "drift" away from those in the encoder if many successive predictions are used, even in the absence of transmission errors. The amount of drift is controlled by requiring that each macroblock be coded without prediction (intra-coded) at least once in any 132 consecutive frames.

The process whereby a decoder becomes initialized or re-initialized with valid picture data without reference to previously transmitted picture information is termed *image refresh*. Image refresh is accomplished by the use of intraframe coded macroblocks. There are two general classes of image refresh which can be used either independently or jointly, periodic transmission of I-frames and progressive refresh.

5.6.1 Periodic transmission of I-frames

One approach to image refresh is to periodically code an entire frame using only intraframe coding. In this case the intra-coded frame is typically an I-frame.[7] The period between successive intra-coded frames may be constant or it may vary.

When a receiver tunes into a new channel where I-frame coding is used for image refresh it may perform the following steps:

- Ignore all data until receipt of the first sequence header
- Decode the sequence header and configure circuits based on sequence parameters
- Ignore all data until the next received I-frame
- Commence picture decoding and presentation

When a receiver processes data which contains uncorrectable errors in an I or P-frame there will typically be a propagation of picture errors due to the use of predictive coding. Pictures received after the error may be decoded incorrectly until an error-free I-frame is received.

5.6.2 Progressive refresh

An alternative method for accomplishing image refresh is to encode only a portion of each picture using the intraframe mode. In this case the intraframe coded regions of

[6] MPEG did not fully specify the results of the IDCT to allow for evolutionary improvements in implementations of this computationally intensive operation.

[7] Note that MPEG allows a field-structured I-frame to consist of a first field which is coded with type I, and a second field which is coded with type P, using the first field as its prediction reference. Although prediction is used within the frame, no reference is made to previously transmitted frames.

each picture should be chosen such that, over the course of a reasonable number of frames, all macroblocks are coded intraframe at least once. In addition constraints might be placed on motion vector values in order to avoid possible contamination of refreshed regions through predictions using unrefreshed regions in an uninitialized decoder.

5.7 Discrete cosine transform (DCT)

Predictive coding in the ATV compression algorithm exploits the temporal correlation in the sequence of image frames. Motion compensation is a refinement of that temporal prediction that allows the coder to account for apparent motions in the image that can be estimated. Aside from temporal prediction, another source of correlation that represents redundancy in the image data is the spatial correlation within an image frame or field. This spatial correlation of images, including parts of images that contain apparent motion, can be accounted for by a spatial transform of the prediction differences. In the intraframe coding case (I-frames), where there is by definition no attempt at prediction, the spatial transform applies to the actual picture data. The effect of the spatial transform is to concentrate a large fraction of the signal energy in a few transform coefficients.

In order to exploit spatial correlation in intraframe and predicted portions of the image, the image prediction residual pixels are represented by their DCT coefficients. For typical images, a large fraction of the energy is concentrated in a few of these coefficients. This makes it possible to code only a few coefficients without seriously affecting the picture quality. The DCT is chosen because it has good energy compaction properties, and in addition, results in real coefficients, and there exist numerous fast computational algorithms for its implementation.

5.7.1 Blocks of 8-by-8 pixels

Theoretically, a DCT of larger size will outperform a DCT of smaller size in terms of coefficient decorrelation and block energy compaction. Better overall performance can be achieved, however, by subdividing the frame into many smaller regions each of which is individually processed. The motivation for this can be understood by the following. If we compute the DCT of the entire frame, we treat the whole frame equally. For a typical image, some regions contain a large amount of detail and other regions contain very little detail. By exploiting the changing characteristics of different images and of different portions of the same image, significant improvements in performance can be realized. In order to take advantage of the varying characteristics of the frame over its spatial extent, the frame is partitioned into blocks of 8-by-8 pixels. The blocks are then independently transformed and adaptively processed based on their local characteristics. The partitioning of the frame into small blocks before taking the transform not only allows spatially adaptive processing, but also reduces the computational and memory requirements. Partitioning the signal into small blocks before computing the DCT is referred to as the Block DCT.

An additional advantage of using the DCT domain representation is that the DCT coefficients contain information about the spatial frequency content of the block. By utilizing the spatial frequency characteristics of the human visual system, the precision with which the DCT coefficients are transmitted can be in accordance with their perceptual importance. This is achieved through the quantization of these coefficients, as explained in the following section.

5.7.2 Adaptive field/frame DCT

As noted above the DCT allows taking advantage of the typically high degree of spatial correlation in natural scenes. When coding interlaced pictures on a frame basis, however, it is possible that significant amounts of motion result in relatively low spatial correlation in some regions. This situation is accommodated by allowing the DCTs to be computed either on a field basis or on a frame basis. The decision to use field or frame-based DCT is made individually for each macroblock.

5.8 Adaptive quantization

The goal of video compression is to maximize the video quality at a given bit rate. This requires a wise distribution of the limited number of available bits. By exploiting the perceptual irrelevancy (as explained in Section 5.7.1) and statistical redundancy (as explained in Section 5.9) within the DCT domain representation, an appropriate bit allocation can yield significant improvements in performance. Quantization is performed to reduce the precision of the DCT coefficient values, and through quantization and codeword assignment, the actual bit rate compression is achieved. The quantization process is the source of virtually all of the loss of information in the compression algorithm. This is important, as it simplifies the design process and facilitates fine tuning of the system.

5.8.1 Adaptive step sizes

The degree of subjective picture degradation caused by coefficient quantization tends to depend on the nature of the scenery being coded. Within a given picture distortions of some regions may be less apparent than in others. The video coding system allows for the level of quantization to be adjusted for each macroblock in order to save bits, where possible, through coarse quantization.

5.8.2 Perceptual weighting

The human visual system is not uniformly sensitive to coefficient quantization error. Perceptual weighting of each source of coefficient quantization error is used to increase quantization coarseness in order to lower the bit rate. The amount of visible distortion resulting from quantization error for a given coefficient depends on the coefficient number, or frequency, the local brightness in the original image, and the duration or the temporal characteristic of the error. DC coefficient error results in mean value distortion for the corresponding block of pels which can expose block boundaries. This is more visible than higher frequency coefficient error which appears as noise or texture.

Displays and human visual systems exhibit non-uniform sensitivity to detail as a function of local average brightness. Loss of detail in dark areas of the picture is not as